Utilising Knowledge Graph Embeddings for Data-to-Text Generation

Nivranshu Pasricha[1], **Mihael Arcan**[2] and **Paul Buitelaar**[1,2]
[1]SFI Centre for Research and Training in Artificial Intelligence
[2] Insight SFI Research Centre for Data Analytics
Data Science Institute, National University of Ireland Galway
n.pasricha1@nuigalway.ie

Abstract

Data-to-text generation has recently seen a move away from modular and pipeline architectures towards end-to-end architectures based on neural networks. In this work, we employ knowledge graph embeddings and explore their utility for end-to-end approaches in a data-to-text generation task. Our experiments show that using knowledge graph embeddings can yield an improvement of up to 2 – 3 BLEU points for *seen* categories on the WebNLG corpus without modifying the underlying neural network architecture.

1 Introduction

Data-to-text generation is concerned with building systems that can produce meaningful texts in a human language from some underlying non-linguistic representation of information (Reiter and Dale, 2000). Traditionally, most applications for data-to-text generation have relied on rule-based systems which are designed using a modular pipeline architecture (Gatt and Krahmer, 2018). However, there has been a shift recently towards end-to-end architectures using neural networks to convert data in the input to text in a natural language in the output. This trend is largely inspired by the success of the end-to-end approaches in the related task of machine translation as well as the availability of large corpora for data-to-text generation such as the WikiBio (Lebret et al., 2016) or the ROTOWIRE (Wiseman et al., 2017) datasets, which contain input data in the form of a table consisting of rows and columns. However, the structure and representation of the input data can vary significantly depending on the task at hand. For example, the input can also be a knowledge graph (KG) represented as a set of RDF-triples like the WebNLG corpus (Gardent et al., 2017) or a dialogue-act-based meaning representation like the E2E dataset (Novikova et al., 2017).

In this work, we employ pre-trained knowledge graph embeddings (KGEs) for data-to-text generation with a model which is trained in an end-to-end fashion using an encoder-decoder style neural network architecture. These embeddings have been shown to be useful in similar end-to-end architectures especially in domain-specific and under-resourced scenarios for machine translation (Moussallem et al., 2019). We focus on the WebNLG corpus which contains RDF-triples paired with verbalisations in English. We compare the use of KGEs to two baseline models – the standard sequence-to-sequence model with attention (Bahdanau et al., 2015) and the transformer model (Vaswani et al., 2017). We also do a comparison with pre-trained GloVe word-embeddings (Pennington et al., 2014).

2 Related Work

Castro Ferreira et al. (2019) have compared pipeline-based and end-to-end architectures for data-to-text generation on the WebNLG corpus. Their findings suggest that the systems which are trained end-to-end are comparable to pipeline methods on seen data categories but do not generalise to new and unseen domains of data. Marcheggiani and Perez-Beltrachini (2018) proposed an encoder based on graph convolutional networks to exploit the structure in the input for an end-to-end system which showed a slight improvement over the standard LSTM encoder. However, their test set did not include data from new and unseen categories. As an alternative to end-to-end training, Moryossef et al. (2019) suggested to split the task into two stages, where the first stage is responsible for text planning while the second stage focuses on text realization. This approach generates fluent text in the output using a neural generation component while at the same time the planning stage gives explicit control over the structure of the generated text.

3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+), Dublin, Ireland (Virtual), 18 December 2020, pages 48–53,

Number of ...	train seen	dev seen	test seen	test unseen
... data-text pairs	20,470	2,550	2,494	1,726
... triples	62,209	7,801	7,527	4897
... entities	1,776	1,142	1,159	863
... lexicalisation tokens	493,236	62,416	60,318	39,389
... lexicalisation types	4,806	2,819	2,795	2,147

Table 1: WebNLG corpus statistics.

200 Public Square	location	Cleveland
200 Public Square	floorCount	45
Cleveland	country	United States
200 Public Square has 45 floors and is located in Cleveland, United States.		
BUILDING	location	LOCATION
BUILDING	floor count	FLOORCOUNT
LOCATION	country	COUNTRY
BUILDING has FLOORCOUNT floors and is located in LOCATION, COUNTRY.		

Table 2: Example of an input tripleset paired with reference text in the output (top) and corresponding delexicalised version (bottom).

Annervaz et al. (2018) classifies KGEs into two categories: structure-based and semantically enriched. Structure-based embeddings encode only entities and relations while semantically-enriched also take into account the associated semantic information. Approaches where relationships are interpreted as displacements operating on the low-dimensional embeddings of the entities, have been implemented within the TransE toolkit (Bordes et al., 2013). RDF2Vec (Ristoski and Paulheim, 2016) uses language modelling approaches for unsupervised feature extraction from sequences of words and adapts them to RDF graphs. Cochez et al. (2017) exploited the Global Vectors algorithm in RDF2Vec to compute embeddings from the co-occurrence matrix of entities and relations. However, Joulin et al. (2017b) showed that a BoW based approach with the *fastText* algorithm (Joulin et al., 2017a) generates state-of-the-art results in KGEs. For data-to-text generation, Chen et al. (2019) have shown that leveraging external knowledge is useful in generating text from Wikipedia infoboxes. In our work, we incorporate pre-trained KGEs based on the *fastText* model with an end-to-end approach for the data-to-text generation.

3 Dataset Description

The WebNLG corpus consists of data units made up of RDF-triples extracted from DBpedia (Auer et al., 2007), paired with reference text lexicalisations. These texts contain sequences of one or more short sentences in English, verbalising the data units in the input. The corpus contains triplesets from 15 DBpedia categories and is divided into two subsets, *seen* and *unseen* for evaluation. The ten *seen* categories are *Airport*, *Astronaut*, *Building*, *City*, *ComicsCharacter*, *Food*, *Monument*, *SportsTeam*, *University* and *WrittenWork* and the five *unseen* categories are *Artist*, *Athlete*, *CelestialBody*, *Company*, *MeanOfTransportation* and *Politician*. The corpus[1] contains 16,095 RDF-triples in the input paired with 42,873 lexicalisations in the output. We follow the same structure for splitting the dataset into training and test sets as defined in the challenge. The final evaluation is done on a test set split into *seen* and *unseen* categories and the training set contains data only from the *seen* categories (Table 1).

[1] https://gitlab.com/shimorina/webnlg-dataset, v2.1

4 Methodology

Knowledge Graph Embeddings To leverage KGEs, we train the embeddings on around 4.2 million entities and 661 relations represented in the entire DBpedia repository, where each subject s, or object o, of an RDF-triple can be associated as a point in a continuous vector space. The relation or the predicate p between the subject and the object entities can be modelled as displacement vector ($s + p = o$) while still preserving the inherent structure of the KG. In this work, we use *fastText* (Joulin et al., 2017b), which is based on a bag-of-words representation and considers the subject s and the object o entities along with the predicate p as a unique discrete token. Thus, *fastText* models co-occurrences of these entities and relations and learns a word representation using a hierarchical softmax. This allows us to create semantically-enriched KGEs which are 500-dimensional vectors that are used to initialise the encoder and decoder embedding layers in our end-to-end system. As a comparison to the usage of KGEs, we engage 300-dimensional GloVe embeddings as pre-trained textual embeddings. For each model, we compare the results with and without a delexicalisation step.

Delexicalisation Before training a model, we perform a delexicalisation step where we modify the RDF-triple in the input similar to Gardent et al. (2017). The subject of the RDF-triple is replaced

	All	Seen	Unseen	All	Seen	Unseen	All		Seen		Unseen	
		BLEU			**METEOR**		**Precision**	**Recall**	**Precision**	**Recall**	**Precision**	**Recall**
LSTM	35.0	53.8	4.3	26.3	39.0	8.2	67.4	49.5	90.0	73.8	37.7	7.0
LSTM + GloVe	38.0*	56.4*	4.2	27.9*	**41.6***	7.3	73.3*	52.6*	94.9*	78.7*	45.9*	6.6
LSTM + KGEs	36.1*	**56.9***	5.1*	27.5*	40.4*	9.2*	70.1*	51.8*	93.6*	75.1*	44.6*	10.2*
Transformer	36.8*	54.6	3.7	27.2	40.7*	7.4	64.8	51.7*	92.6*	77.3*	31.1	7.1
Transformer + GloVe	39.0*	54.9*	6.9*	28.1*	40.7*	8.9	66.3	52.2*	92.6*	76.4*	36.6	9.4
Transformer + KGEs	38.8*	56.4*	5.3*	27.6*	40.7*	8.2	70.8*	51.2*	94.6*	74.8*	42.0*	9.7*
DELEX												
LSTM	42.0	52.4	15.2	28.8	39.9	10.9	85.9	54.9	94.3	77.5	**86.3**	15.5
LSTM + GloVe	**44.5***	56.1*	11.9	28.6	40.6	9.5	88.3*	54.4	96.5*	**79.2***	72.9	12.2
LSTM + KGEs	43.0*	54.5*	14.8	**28.8**	39.8	11.5	88.4*	55.1	96.1*	77.7	75.8	16.2
Transformer	38.0	50.9	11.5	27.9	38.8	10.8	83.3	54.0	94.2	76.5	77.4	15.4
Transformer + GloVe	44.2*	55.6*	**16.7***	28.6	39.2	**11.7**	89.6*	**54.9**	96.3*	75.3	80.1	**19.8***
Transformer + KGEs	43.2*	54.1*	15.8*	28.5	39.4	11.2	**89.9***	54.2	**96.9***	77.5	77.0	15.4

Table 3: Results for *all*, *seen* and *unseen* categories. **DELEX** indicates models trained on delexicalised corpus. Models marked with $*$ are significantly better than the baseline LSTM model ($p < 0.05$). The best performing model in each category is highlighted in bold face.

by the DBpedia category and the object is replaced by the predicate using a pre-defined delexicalisation dictionary[2]. We also split the predicate in each triple on the *camelCase* using a regular expression. We perform this delexicalisation step on the reference text too based on the corresponding entities in the input, as shown in Table 2.

5 Experimental Setup

We follow the WebNLG baseline system (Gardent et al., 2017) as one of the baseline architectures for our experiments, which is a vanilla sequence-to-sequence LSTM model with attention where the RDF triples in the input are linearised as a sequence and the output text is tokenised before training. We use another baseline based on the transformer architecture similar to the end-to-end architecture setup by Castro Ferreira et al. (2019).

The models are trained using the OpenNMT library (Klein et al., 2017). We use the default parameters for the baseline model: two hidden layers with 500 LSTM units per hidden layer and word embeddings of 500 dimensions. Dropout is applied with value 0.3 and the LSTM models are trained with stochastic gradient descent, starting with a learning rate of 1.0 and learning rate decay enabled. For the transformer model, the encoder-decoder setup contains 6 layers with 512 hidden units. The word embeddings are 512-dimensional and the feed-forward sublayers are 2048-dimensional. Each multi-head attention sublayer consists of 8 attention heads. Dropout is applied with value 0.1 and the model is trained using Adam optimizer (Kingma and Ba, 2015).

The overlap between the model vocabulary and GloVe embeddings is 44.36% while for KGEs it is 45.66%. For the delexicalised corpus, the overlap increases to 46.70% for GloVe and 47.40% for KGEs respectively. Since the size of the vocabulary of the training corpus is limited to a few thousand words (around 5,000 unique words), we enable the options for dynamic dictionary and shared vocabulary to allow the model to share tokens on between the source and the target side. All models are trained for 100,000 steps and for evaluation on the test set we take the checkpoint with the best BLEU score on the validation set.

6 Results and Discussion

In this section, we report the results of our experiments in terms of two commonly used evaluation metrics, BLEU (Papineni et al., 2002) and METEOR (Denkowski and Lavie, 2014). We also report scores in terms of precision and recall of the input entities covered in the output generations. For statistical significance, we use MultEval (Clark et al., 2011) to perform bootstrap resampling and report the results on three test sets consisting of instances from *seen*, *unseen* and *all* categories in Table 3.

For the test set consisting of *seen* categories, using KGEs shows consistent improvement over the baseline models for both LSTM and transformer architectures. This improvement is observed in both cases whether the delexicalisation step is performed or not. For the models trained on the delex-

[2] https://gitlab.com/webnlg/webnlg-baseline/

Input Triples	Acta Mathematica Hungarica	LCCN number	83646315
	Acta Mathematica Hungarica	abbreviation	Acta Math . Hungar .
	Acta Mathematica Hungarica	academicDiscipline	Mathematics
	Acta Mathematica Hungarica	ISSN number	0236 - 5294

Reference	Acta Mathematica Hungarica (abbreviated as " Acta Math . Hungar . ") covers the academic discipline of Mathematics . It has the ISSN number of " 0236 - 5294 " and LCCN number 83646315 .

Model	**Generation**	**BLEU**	**METEOR**
LSTM	Acta Mathematica Hungarica (Acta Math . Hungar .) which *has a discipline of math* has the LCCN number 83646315 and ISSN number **1588 - 2632** .	27.67	32.24
LSTM + GloVe	Acta Mathematica Hungarica is abbreviated to Acta Math . Hungar . It has the LCCN number 83646315 and the ISSN number 0236 - 5294 and LCCN number 83646315 .	58.98	40.65
LSTM + KGEs	Acta Mathematica Hungarica has the abbreviation of " Acta Math . Hungar . " and *comes under the academic discipline of Math* . It has the LCCN number 83646315 and the ISSN number 0236 - 5294 .	53.31	44.96
Transformer	Acta Mathematica Hungarica , or Acta Math . Hungar . has a LCCN number of 83646315 and a ISSN number of 0236 - 5294 .	34.75	39.75
Transformer + GloVe	Acta Mathematica Hungarica , or Acta Math . Hungar . has a LCCN number of 83646315 and a ISSN number of 0236 - 5294 .	34.75	39.75
Transformer + KGEs	Acta Mathematica Hungarica is abbreviated to Acta Math . Hungar . It has the LCCN number 83646315 and the ISSN number 0236 - 5294 .	55.26	42.47

Table 4: Qualitative example highlighting differences in the generations produced the different models. Mistakes are highlighted in bold face, while fragments with underline correspond to the *abbreviation* predicate in the second triple and fragments in italics refer to the *academicDiscipline* predicate in the third input triple.

icalised corpus, GloVe embeddings appear to produce better results than KGEs. This is due to the fact that the KGEs are not trained on delexicalised entities. Nonetheless, we observe delexicalisation to be useful in generating text for *unseen* categories where almost none of the entities and properties are present in the training set. Our results are consistent with Gardent et al. (2017) and Castro Ferreira et al. (2019), who have also shown that end-to-end neural approaches perform well on *seen* categories but struggle to generalise on the *unseen* ones.

Similar to Wiseman et al. (2017), we propose entity-based evaluation metrics to measure the coverage of subject and object entities in the output generation. For each subject and object entity in the triples, we count unique entities extracted from the output text, and calculate precision and recall scores. A drawback of this approach is that it does not penalise multiple repetitions of the same entity in the output. However, it is useful as a measure of how well the entities are represented in the output. Our results suggest both KGEs and GloVe embeddings yield an improvement in precision and recall, especially in the *seen* categories. Delexicalisation also leads to an overall improvement in the scores, however, neither KGEs nor GloVe embeddings appear to do well on *unseen* categories for the delexicalised corpus in terms of these metrics.

Usually, data-to-text generation tasks involve only non-linguistic entities in the input, which makes it difficult to generate fluent and coherent text in the output without explicitly defining the rules for a mapping between the input entities and output text. However, in the WebNLG corpus, the predicate inside each triple can be considered as a linguistic entity around which the output sentence is structured. Table 4 shows a qualitative example where the output from each model is structured slightly differently on the two input predicates *abbreviation* and *academicDiscipline*. This example also highlights some of the mistakes and errors produced by neural text generation approaches. For instance, the baseline model produces incorrect ISSN number in the output, while the LSTM+GloVe model repeats the LCCN number twice. Another notable error is the omission of *academicDiscipline* in the output by all models except the baseline LSTM and LSTM+KGEs models, which highlights the difficuly in generating text from a large number of triples in the input.

7 Conclusion and Future Work

In this work we showed that using KGEs to initialise the encoder and decoder embeddings in neural data-to-text generation consistently achieves better performance by up to 2 or 3 BLEU points over the baseline models for the *seen* categories on the WebNLG corpus. For future work, we plan to leverage KGEs with delexicalised DBpedia entries, specifically to target the *unseen* data.

Acknowledgments

This work was conducted with the financial support of the Science Foundation Ireland Centre for Research Training in Artificial Intelligence under Grant No. 18/CRT/6223 and co-supported by Science Foundation Ireland under grant number SFI/12/RC/2289 2 (Insight), co-funded by the European Regional Development Fund.

References

KM Annervaz, Somnath Basu Roy Chowdhury, and Ambedkar Dukkipati. 2018. Learning beyond datasets: Knowledge graph augmented neural networks for natural language processing. In *Proceedings of the 2018 Conference of NAACL-HLT*, pages 313–322. ACL.

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: A nucleus for a web of open data. In *The semantic web*, pages 722–735. Springer.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. *Proceedings of the Third International Conference on Learning Representations*.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems*, pages 2787–2795.

Thiago Castro Ferreira, Chris van der Lee, Emiel van Miltenburg, and Emiel Krahmer. 2019. Neural data-to-text generation: A comparison between pipeline and end-to-end architectures. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 552–562, Hong Kong, China. Association for Computational Linguistics.

Shuang Chen, Jinpeng Wang, Xiaocheng Feng, Feng Jiang, Bing Qin, and Chin-Yew Lin. 2019. Enhancing neural data-to-text generation models with external background knowledge. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3022–3032, Hong Kong, China. Association for Computational Linguistics.

Jonathan H. Clark, Chris Dyer, Alon Lavie, and Noah A. Smith. 2011. Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 176–181, Portland, Oregon, USA. Association for Computational Linguistics.

Michael Cochez, Petar Ristoski, Simone Paolo Ponzetto, and Heiko Paulheim. 2017. Global rdf vector space embeddings. In *International Semantic Web Conference*, pages 190–207. Springer.

Michael Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 376–380, Baltimore, Maryland, USA. Association for Computational Linguistics.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. The WebNLG Challenge: Generating Text from RDF Data. In *Proceedings of the 10th International Conference on Natural Language Generation*, pages 124–133, Santiago de Compostela, Spain. Association for Computational Linguistics.

Albert Gatt and Emiel Krahmer. 2018. Survey of the State of the Art in Natural Language Generation: Core tasks, applications and evaluation. *Journal of Artificial Intelligence Research*, 61:65–170.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017a. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Maximilian Nickel, and Tomas Mikolov. 2017b. Fast linear model for knowledge graph embeddings. In *The NIPS 2017 AKBC Workshop*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, Vancouver, Canada. Association for Computational Linguistics.

Rémi Lebret, David Grangier, and Michael Auli. 2016. Neural Text Generation from Structured Data with Application to the Biography Domain. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1203–1213, Austin, Texas. Association for Computational Linguistics.

Diego Marcheggiani and Laura Perez-Beltrachini. 2018. Deep graph convolutional encoders for structured data to text generation. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 1–9, Tilburg University, The Netherlands. Association for Computational Linguistics.

Amit Moryossef, Yoav Goldberg, and Ido Dagan. 2019. Step-by-step: Separating planning from realization in neural data-to-text generation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2267–2277, Minneapolis, Minnesota. Association for Computational Linguistics.

Diego Moussallem, Axel-Cyrille Ngonga Ngomo, Paul Buitelaar, and Mihael Arcan. 2019. Utilizing knowledge graphs for neural machine translation augmentation. In *Proceedings of the 10th International Conference on Knowledge Capture*, K-CAP '19, page 139–146, New York, NY, USA. Association for Computing Machinery.

Jekaterina Novikova, Ondřej Dušek, and Verena Rieser. 2017. The E2E dataset: New challenges for end-to-end generation. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 201–206, Saarbrücken, Germany. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Ehud Reiter and Robert Dale. 2000. *Building Natural Language Generation Systems*, 1 edition. Cambridge University Press.

Petar Ristoski and Heiko Paulheim. 2016. Rdf2vec: Rdf graph embeddings for data mining. In *International Semantic Web Conference*, pages 498–514. Springer.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, \Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Sam Wiseman, Stuart Shieber, and Alexander Rush. 2017. Challenges in Data-to-Document Generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2253–2263, Copenhagen, Denmark. Association for Computational Linguistics.